CLAIMS

1. A computer-implemented system for designing integrated circuits using reusable design components comprising:

multiple component type definitions corresponding to different types of reusable components, the component type definitions including attributes associated with the component type;

multiple components, each component being of one of the component types;

multiple component versions for each of the multiple components, each component version being of one of the component, at least one of the component versions including;

metadata including a component version number for tracking different versions of the component and attributes values describing attributes of the component; and

a deliverable version including one or more design files performing a common function and having described by deliverable version number.

2. The system of claim 1 further comprising metadata associating the deliverable version number with the component version number.

3. The system of claim 1 in which the metadata corresponding to the component versions are stored in a database different from the database in which the deliverable files are stored.

4. The system of claim 3 further comprising an application server through which the user accesses the database to create and modify the metadata database.

-37-

5. The system of claim 1 further comprising multiple deliverable type definitions corresponding to different types of deliverables.

6. The system of claim 5 in which at least one of the deliverable types having associated qualification requirements defining requirements to qualify deliverable versions of that type to a qualification level.

7. The system of claim 6 in which at least one of the deliverable types have multiple associated qualification requirements defining requirements to qualify deliverable versions of that type to multiple qualification levels.

8. The system of claim 1 further comprising a design flow template associated with the component type, the design flow template specifying design tasks for designing a component of the component type.

9. The system of claim 8 further comprising a design flow version that is an instantiation of the design flow template and that correspond to the component version.

10. The system of claim 1 further comprising information for tracking design problems connected with the component version, the information including a person responsible for fixing each problem and the steps taken or required to be taken to fix each problem.

11. The system of claim 1 in which at least one of the component versions instantiates a second component version of a second component type, the second component version including at least one second deliverable version.

12. The system of claim 11 in which the component type corresponds to a system on a chip.

13. The system of claim 12 in which the second component type corresponds to a reusable design block.

14. The system of claim 1 further comprising users names, at least some of the users names are grouped into teams.

15. The system of claim 14 in which the users and teams have selectively controllable access to create, view or change components and to grant other users the right to create, view or change components.

16. The system of claim 14 in which the users and teams have selectively controllable access to create, view or change component versions and to grant other users the right to create, view or change component versions.

17. The system of claim 16 in which access to create, view or change component versions and to grant other users the right to create, view or change component versions is inherited from a team to its members.

18. The system of claim 1 further comprising a communications system in which a user can a specify a type of design event of interest and the user is automatically notified of the occurrence of design events of that type or in which a user can specify that other users to be notified of the occurrence of a design event.

19. The system of claim 18 in which the design event is a qualification of a component version.

20. The system of claim 18 in which the communications system includes a public mailboxes to receive information of public interest and private mail boxes associated with

individual users to receive design event information which the individual user has request or that another individual user has request be sent to the first individual user.

21. The system of claim 1 further comprising metadata describing the origins of the component version, the origin metadata including ancestry and hierarchy information to allow for tracking of shared design information.

22. A method of designing integrate circuits comprising:

defining deliverable types;

defining a first component type including specifying attributes and ones of the deliverable types to be associate with components of the first component type;

defining qualification standards associated with the first component type;

defining qualification questions associated with the deliverable types;

creating a first component of the first component type;

creating a first component version of the first component;

creating a deliverable version of the deliverable types associated with the first component type;

qualifying the deliverable version; and

qualifying the first component version.

23. The method of claim 22 further comprising:

defining a second component type including specifying attributes and ones of the deliverable types to be associate with components of the second component type;

defining qualification standards associated with the second component type;

creating a component version of the second component type; and

-40-

instantiating the first component version as part of the second component

version, thereby reusing the first component version.

24. The method of claim 23 further comprising maintaining information in connection

with the second component that describes the hierarchy of the first component.

25. The method of claim 22 further comprising creating a flow comprising a task list

associated with the creating a deliverable version of the deliverable type, the task list

showing the required input and output of each task where each input and output is associated

with a deliverable type.